This Maple worksheet accompanies the papers:

Di Nardo E., G. Guarino, D. Senato (2008), *A Maple algorithm for polykays and their generalizations*, Adv. Appl. Stat. Vol. 8, No. 1, 19 - 36, http://www.pphmj.com/journals/adas.htm.

Di Nardo E., G. Guarino, D. Senato (2008), *An unifying framework for k-statistics, polykays and their generalizations,* Bernoulli. Vol. 14(2), 440-468. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability, http://isi.cbs.nl/bernoulli/, (download from http://www.unibas.it/utenti/dinardo/lavori.html)

Di Nardo E., G. Guarino, D. Senato (2008), *Symbolic computation of moments of sampling distributions,* Comp. Stat. Data Analysis Vol. 52, no. 11, 4909-4922, (download from http://arxiv.org/PS_cache/arxiv/pdf/0806/0806.0129v1.pdf or http://www.unibas.it/utenti/dinardo/lavori.html)

Multiset subdivision

E. Di Nardo*
elvira.dinardo@unibas.it
http://www.unibas.it/utenti/dinardo/home.html;
Tel: +39 0971205890, Fax: +39 0971205896

G. Guarino**
giuseppe.guarino@asl2.potenza.it

D. Senato*
domenico.senato@unibas.it

* Dipartimento di Matematica e Informatica, Università degli Studi della Basilicata,Viale dell'Ateneo Lucano n.10, 85100 Potenza, Italy

**Medical Scool, Università del Sacro Cuore (Rome branch), Largo Agostino Gemelli n.8, 00168 Roma, Italy

Introduction

Abstract: .The algorithm allows us to build subdivisions of multiset, reducing the overall computational complexity using the integer partitions.
Application Areas/Subject: Combinatorics & algebraic and combinatorial methods
Keyword: Set partitions, multiset
See Also: For applications see [3]

Initialization

> *restart*

> *with*(*combinat*, *partition*, *multinomial*);

$$[partition, multinomial] \tag{2.1}$$

Background to Subdivisions of a Multiset

Consruction of set partitions

List all partitions of a set. The parameter "N" indicates the number of blocks in which the set is subdivided.

```
setParts := proc(N)
    local b, listParts;
    if N = 0 then return {{ }} end if;
    listParts := {{α_1}};
    for b in [seq(α_x, x = 2..N)] do
        listParts := seq(op({seq({op(`minus`(x, {y})), {op(y), b}}, y = x), {op(x), {b}}}), x
         = {listParts})
    end do;
    listParts
end proc:
```

Example

> *setParts*(3)

$$\{\{\alpha_1, \alpha_2, \alpha_3\}\}, \{\{\alpha_3\}, \{\alpha_1, \alpha_2\}\}, \{\{\alpha_1\}, \{\alpha_2, \alpha_3\}\}, \{\{\alpha_2\}, \{\alpha_1, \alpha_3\}\}, \{\{\alpha_1\}, \{\alpha_2\}, \{\alpha_3\}\} \tag{3.1.1}$$

Subdivision of a special multiset: $M = \{\alpha^{(i)}\}$

When the multiset has support equal to a $\{\alpha\}$, the output is the follow:

> $vEval := [seq(\alpha_i = \alpha, i = 1..3)]$;
$L := seq([seq(mul(z, z = y), y = x)], x = [setParts(3)])$;
$Partitions = eval([L], vEval)$

$$vEval := [\alpha_1 = \alpha, \alpha_2 = \alpha, \alpha_3 = \alpha]$$

$$L := [\alpha_1\, \alpha_2\, \alpha_3], [\alpha_3, \alpha_1\, \alpha_2], [\alpha_1, \alpha_2\, \alpha_3], [\alpha_2, \alpha_1\, \alpha_3], [\alpha_1, \alpha_2, \alpha_3]$$

$$Partitions = [[\alpha^3], [\alpha, \alpha^2], [\alpha, \alpha^2], [\alpha, \alpha^2], [\alpha, \alpha, \alpha]] \tag{3.2.1}$$

The output can be compacted by using the following notation where the subdivisions appear togeter with their multiplicity:

$$[[[\alpha, \alpha, \alpha], 1], [[\alpha, \alpha^2], 3], [[\alpha^3], 1]].$$

The above print-out is the output of the following routine:

```
> partInt :=proc(nInt)
      local vEval;
      vEval := [seq(i = α^i, i = 1 ..nInt)];
      [seq([eval(y, vEval), nInt! / mul(x!^numboccur(y, x) ·numboccur(y, x)!, x = {op(y)})], y
       = partition(nInt))]
   end proc:
```

Example

```
> partInt(3)
```

$$[[[\alpha, \alpha, \alpha], 1], [[\alpha, \alpha^2], 3], [[\alpha^3], 1]] \tag{3.2.2}$$

Subdivisions of a multiset

When in the multiset there are two or more different elements, the result is different from the previous one. Also in this case we use integer partitions for constructing subdivisions. This device reduces the computational complexity.
The list of all partitions of a set with 3 blocks is:

$$\{\{\alpha_1, \alpha_2, \alpha_3\}\}, \{\{\alpha_1, \alpha_2\}, \{\alpha_3\}\}, \{\{\alpha_2\}, \{\alpha_1, \alpha_3\}\}, \{\{\alpha_1\}, \{\alpha_2, \alpha_3\}\}, \{\{\alpha_1\}, \{\alpha_2\}, \{\alpha_3\}\}.$$

Setting $[\alpha_1 = \alpha, \alpha_2 = \alpha, \alpha_3 = \beta]$ we obtain:

$$[[\alpha^2 \beta], [\alpha^2, \beta], [\alpha, \alpha, \beta], [\alpha, \alpha \beta], [\alpha, \alpha \beta]]$$

Compacting the previous output we obtain:

$$[[[\alpha \beta, \alpha], 2], [[\alpha, \alpha, \beta], 1], [[\alpha^2 \beta], 1], [[\alpha^2, \beta], 1]]$$

The Maple routines

Some details on secondary Maple routines

The following function calcules the product of factorials of multiplicity.

```
> nRep :=proc(u) mul(x_2!, x = convert(u, multiset)) end proc:
```

Example

```
> nRep([a, a, a, b, b])
```

$$12 \tag{4.1.1.1}$$

The following function inserts an element into an array with the following rules:
1) the element must not be in a previous block with the same or less degree
2) jump every block equal to the previous

```
> URv :=proc(u, v)
      local U, ou, i, ptr, vI;
      ou := NULL; U := [ ]; vI := indets(v);
      for ptr from nops(u) by −1 to 2 do
         if has(u_ptr, v) then break end if
      end do;
```

for i *from* ptr *to* $nops(u)$ *do*
if not $\left(u_i = ou \text{ or } has\left(u_i, vI\right)\right)$ *then*
$ou := u_i;$
$U := \left[op(U), \left[op\left(u_{1..i-1}\right), u_i \cdot v, op\left(u_{i+1..-1}\right)\right]\right]$
end if
end do;
$op(U), [op(u), v]$
end proc:

Examples

> $URv([a, a, a], b)$

$$[a\, b, a, a], [a, a, a, b] \quad \textbf{(4.1.1.2)}$$

> $URv([a, a, b, a], b)$

$$[a, a, b, a\, b], [a, a, b, a, b] \quad \textbf{(4.1.1.3)}$$

> $URv([a, a, a, b], b)$

$$[a, a, a, b, b] \quad \textbf{(4.1.1.4)}$$

> $URv\left(\left[a, a, a, b^2\right], b\right)$

$$\left[a, a, a, b^2, b\right] \quad \textbf{(4.1.1.5)}$$

> $URv\left([a, a, a, b], b^2\right)$

$$\left[a\, b^2, a, a, b\right], \left[a, a, a, b, b^2\right] \quad \textbf{(4.1.1.6)}$$

>

The following function inserts more then one elements into an array with the same rules.
Note: the array contains also the multeplicity of the multiset

> $URV :=$ ***proc***( )
local $U, V, i;$
$U := \left[args_{1,1}\right];\ V := args_{2,1};$
for i *to* $nops(V)$ *do* $U := \left[seq\left(URv\left(u, V_i\right), u = U\right)\right]$ *end do;*
$seq\left(\left[x, args_{1,2} \cdot args_{2,2}\right], x = U\right)$
end proc:

Examples

> $URV([[a, a, a], 1], [[b, b], 1])$

$$[[a\, b, a\, b, a], 1], [[a\, b, a, a, b], 1], [[a, a, a, b, b], 1] \quad \textbf{(4.1.1.7)}$$

> $URV([[a, a, a], 2], [[b, b], 3])$

$$[[a\, b, a\, b, a], 6], [[a\, b, a, a, b], 6], [[a, a, a, b, b], 6] \quad \textbf{(4.1.1.8)}$$

The following function is an extension of previous one. The input can be more then two vectors.

> $URmV :=$ ***proc***( )
local $U, i;$
if $nargs = 1$ *then* $args$ *else*

$U := URV(args_1, args_2);$

for i ***from*** 3 ***to*** $nargs$ ***do*** $U := seq(URV(u, args_i), u = [U])$ ***end do;***

$seq\left(\left[x_1, \frac{x_2}{nRep(x_1)}\right], x = U\right)$

end if

end proc :

Examples

> $URmV([[a, a], 2], [[b], 1], [[c], 1])$

$[[a\, b\, c, a], 2], [[a\, b, a\, c], 2], [[a\, b, a, c], 2], [[a\, c, a, b], 2], [[a, a, b\, c], 1], [[a, a, b, c], 1]$ **(4.1.1.9)**

> $URmV([[a, a], 2], [[a, b], 2], [[b], 3], [[c], 1])$

$[[a\, b\, c, a\, b, a], 12], [[a\, b, a\, b, a\, c], 6], [[a\, b, a\, b, a, c], 6], [[a\, b\, c, a, a, b], 6], [[a\, b, a\, c, a, b], 12], [[a\, b, a, a, b\, c], 6], [[a\, b, a, a, b, c], 6], [[a\, c, a, a, b, b], 3], [[a, a, a, b\, c, b], 2], [[a, a, a, b, b, c], 1]$ **(4.1.1.10)**

The following function makes the Cartesian Product.

> $comb :=$ ***proc***(V, ptr, Y)

if $ptr = nops(V) + 1$ ***then return*** Y ***end if;***

$seq(comb(V, ptr + 1, [op(Y), L]), L = V_{ptr})$

end proc:

Examples

> $comb([[X, Y], [\alpha, \beta, \gamma]], 1, [\ ])$

$[X, \alpha], [X, \beta], [X, \gamma], [Y, \alpha], [Y, \beta], [Y, \gamma]$ **(4.1.1.11)**

> $comb([[X, Y], [\alpha, \beta], [r, s]], 1, [\ ])$

$[X, \alpha, r], [X, \alpha, s], [X, \beta, r], [X, \beta, s], [Y, \alpha, r], [Y, \alpha, s], [Y, \beta, r], [Y, \beta, s]$ **(4.1.1.12)**

The master function

A Maple algorithm for listing multisets subdivisions.

> $makeTab :=$ ***proc***$(\ )$

local $U;$

$U := \left[seq\left(\left[seq\left(\left[\left[seq\left(\alpha_i^z, z = y\right)\right], multinomial\left(args_i, seq(r, r = y)\right)\right], y = partition\left(args_i\right)\right)\right], i = 1..nargs\right)\right];$

$$\textbf{if } nops(U) = 1 \textbf{ then } \left[seq\left(\left[x_1, \frac{x_2}{nRep(x_1)}\right], x = op(U)\right)\right]$$
$$\textbf{else } [seq(URmV(op(x)), x = [comb(U, 1, [\ ])])] \textbf{ end if}$$

end proc:

Examples

> *makeTab*(4)

$$[[[\alpha_1, \alpha_1, \alpha_1, \alpha_1], 1], [[\alpha_1, \alpha_1, \alpha_1^2], 6], [[\alpha_1^2, \alpha_1^2], 3], [[\alpha_1, \alpha_1^3], 4], [[\alpha_1^4], 1]] \quad \textbf{(4.1.2.1)}$$

> *makeTab*(2, 2)

$$[[[\alpha_1\,\alpha_2, \alpha_1\,\alpha_2], 2], [[\alpha_1\,\alpha_2, \alpha_1, \alpha_2], 4], [[\alpha_1, \alpha_1, \alpha_2, \alpha_2], 1], [[\alpha_1\,\alpha_2^2, \alpha_1], 2], \quad \textbf{(4.1.2.2)}$$
$$[[\alpha_1, \alpha_1, \alpha_2^2], 1], [[\alpha_1^2\,\alpha_2, \alpha_2], 2], [[\alpha_1^2, \alpha_2, \alpha_2], 1], [[\alpha_1^2\,\alpha_2^2], 1], [[\alpha_1^2, \alpha_2^2], 1]]$$

> *makeTab*(2, 1, 1)

$$[[[\alpha_1\,\alpha_2\,\alpha_3, \alpha_1], 2], [[\alpha_1\,\alpha_2, \alpha_1\,\alpha_3], 2], [[\alpha_1\,\alpha_2, \alpha_1, \alpha_3], 2], [[\alpha_1\,\alpha_3, \alpha_1, \alpha_2], 2], \quad \textbf{(4.1.2.3)}$$
$$[[\alpha_1, \alpha_1, \alpha_2\,\alpha_3], 1], [[\alpha_1, \alpha_1, \alpha_2, \alpha_3], 1], [[\alpha_1^2\,\alpha_2\,\alpha_3], 1], [[\alpha_1^2\,\alpha_2, \alpha_3], 1], [[$$
$$\alpha_1^2\,\alpha_3, \alpha_2], 1], [[\alpha_1^2, \alpha_2\,\alpha_3], 1], [[\alpha_1^2, \alpha_2, \alpha_3], 1]]$$

Note: the notion of subdivision as been introduced in [2].
In the subdivision list, the sum of all multiplicity is the Bell Number.
Moreover the same multiplicities of the subdivision with the same block numbers is the Stirling Number of the second kind.
Example: from "makeTab(4)" we obtain:

$$[[[\alpha_1, \alpha_1, \alpha_1, \alpha_1], 1], [[\alpha_1, \alpha_1, \alpha_1^2], 6], [[\alpha_1^2, \alpha_1^2], 3], [[\alpha_1, \alpha_1^3], 4], [[\alpha_1^4], 1]]$$

in wich it is possible to observe:
1 = combinat['stirling2'](4,4) =combinat['stirling2'](4,1);
6 = combinat['stirling2'](4,3);
3 + 4 = 7 = combinat['stirling2'](4,2);
and 1 + 6 + 3 + 4 + 1 = 15 = combinat['Bell'](4).

Conclusions

The multiset subdivision is very usefull in speed-up the algorithm that generates power sum from augumented symmetric functions and vice cersa. So it can be used also in the more general theory of simmetric functions. We ave used the multiset subdivision for calcolate k-statistics, polykays and their generalizations (see references).

References

[1] Di Nardo E., G. Guarino, D. Senato (2008) A Maple algorithm for polykays and their

generalizations. Adv. Appl. Stat. Vol. 8, No. 1, 19 - 36, http://www.pphmj.com/journals/adas.htm.

[2] Di Nardo E., G. Guarino, D. Senato (2008) An unifying framework for k-statistics, polykays and their generalizations. Bernoulli. Vol. 14(2), 440-468. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability, http://isi.cbs.nl/bernoulli/, (download from http://www.unibas.it/utenti/dinardo/lavori.html)

[3] Di Nardo E., G. Guarino, D. Senato, *A Maple algorithm for k-statistics, polykays and their multivariate generalization,* source Maple algorithm located in www.maplesoft.com *(submitted)*

[4] Di Nardo E., G. Guarino, D. Senato (2008) Symbolic computation of moments of sampling distributions. Comp. Stat. Data Analysis Vol. 52, no. 11, 4909-4922, (download from http://arxiv.org/PS_cache/arxiv/pdf/0806/0806.0129v1.pdf or http://www.unibas.it/utenti/dinardo/lavori.html)

>